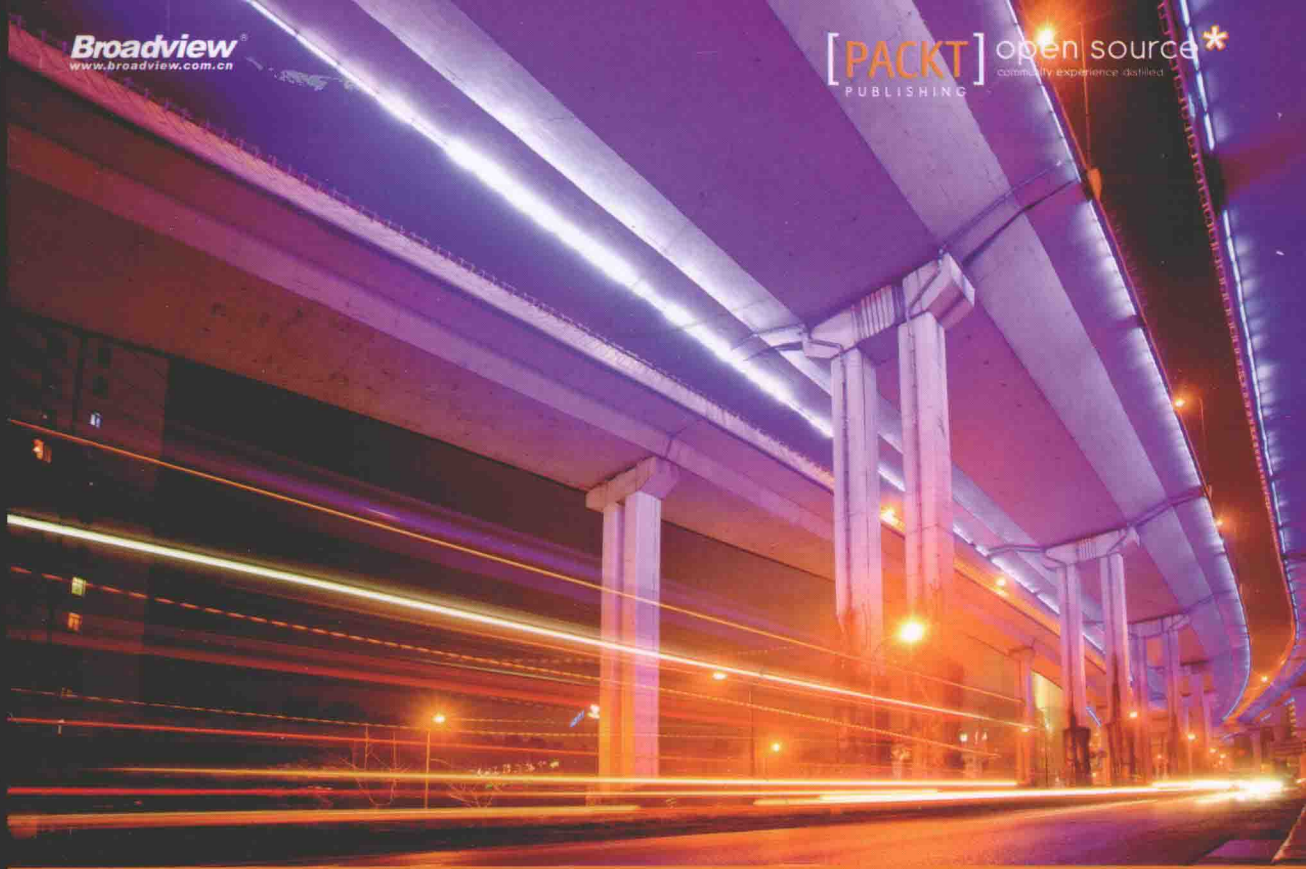

Broadview
www.broadview.com.cn

[PACKT] open source*
PUBLISHING
community experience distilled

Practical DevOps

DevOps 实践

驭DevOps之力强化技术栈并优化IT运行

[瑞典]Joakim Verona 著
高清华 马博文 译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Practical DevOps

DevOps实践

[瑞典]Joakim Verona 著

高清华 马博文 译

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

《DevOps 实践》介绍了 DevOps 的起源和概览，并通过一个贯穿全书的例子，从架构开始，到代码的存储、构建、测试、部署、监控，直至流程的跟踪，推荐了许多可用的工具和可行的示范，是一本 DevOps 实践方面不可多得的参考书籍。

本书面向愿意承担更大责任的开发人员和系统管理员，也很适合愿意更好地支持开发人员的运维人员。无须任何 DevOps 知识即可快速上手！

版权贸易合同登记号　图字：01-2016-3999

图书在版编目（CIP）数据

DevOps 实践 /（瑞典）约阿基姆·维罗纳（Joakim Verona）著；高清华，马博文译.
北京：电子工业出版社，2016.10
书名原文：Practical DevOps
ISBN 978-7-121-29812-7

Ⅰ.①D… Ⅱ.①约… ②高… ③马… Ⅲ.①虚拟处理机 Ⅳ.①TP338

中国版本图书馆 CIP 数据核字(2016)第 207459 号

责任编辑：张春雨
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×980　1/16　印张：13.5　字数：302.4 千字
版　　次：2016 年 10 月第 1 版
印　　次：2016 年 10 月第 1 次印刷
定　　价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819　faq@phei.com.cn。

译者序

什么是 DevOps？我的前同事李光磊将其译为开发自运维，他还写了篇很有意思的博客来说明：http://liguanglei.name/blogs/2015/04/22/devops-chinese-name/。这个将开发和运维结合起来的词，代表了一种文化，那就是大家共同协作。狭义上的大家，指的是开发和运维，广义上，指的是所有软件生命周期里参与的角色。

“共同协作”是个富有正能量的词。感觉上，随便往哪儿一套都是正确的。那为什么要在 DevOps 里着重强调呢？DevOps 到底解决了什么问题？归根结底，就是提高产品质量。爱思考的你，可能心里已经有千万个提高产品质量的方案从脑海里呼啸而过——代码审查、自动化测试、持续集成、代码质量管理工具、程序员鼓励师……对对对，这些方案都能在某种程度上解决一些层次的问题。但是，产品质量的根源在哪儿呢？在于人。如果开发者对自己要做的事情不负责，甚至压根儿不知道后果，怎么能指望这样的人能够生产出来高质量的代码呢？举个例子：作为开发者，我知道自己写的代码会由测试部门来进一步测试，在有进度压力的时候，我就会更倾向于去想：“那就先这么凑合着吧，反正有问题 QA 们会说的”。如果我不知道部署和维护产品是怎么一回事，我就不会主动地在产品里写上日志的代码。对于运维人员来说，由于处于软件生命周期的下游，相信对类似的场景感触更甚。DevOps 能够做到的事，就是让人有这个意识：需要对产品的质量负责。DevOps 能够提供一个平台或机制，让我能够从中找到所需的资源。

“共同协作”也是个虚无缥缈的词。它应该如何落地呢？这就是本书想要给读者们带来的内容。在实践上，从架构开始，到代码的存储、构建、测试、部署、监控，直至流程的跟踪，本书推荐了许多可用的工具和练习，确实无愧于《DevOps 实践》之名。细读全书可以对其有一个全局的概览并充实自己的 DevOps 工具箱；而在实际场景中再查阅本书，将其当作一本各种技术的快速参考手册也不失为明智之举。本书的许多实例通过 Docker 启动，在紧随潮流技术的同时也简化了练习步骤，值得花些时间试试。在企业里，使用自动化和持续交付来提高代码部署频率、降低代码上线间隔。这样的指标是比较容易统计的，在让管理人员满意的同时，也能减少开发和运维的痛苦。只有让各角色都真切地感受到实

惠，大家才会更加愿意从心底接受并积极参与到这一过程中。

“共同协作”是个看上去很美的词。为什么大家还不赶紧拥抱它？因为它的成本可能还挺高的。大型企业在管理上，通常权责分明，从而导致每个角色的成员都不愿意轻易踏足其他领域；流程烦琐，导致一个小小的改进也需要漫长的批复；安全性要求高，引发各种违规，进一步导致没有和其他人分享的意愿；员工操作权限管理精密，上不了网、下不了包、开不了虚拟机……这些问题，虽然不至于疾在骨髓，但起码也在肠胃了。而且，自动化测试、部署流水线等都需要比较高的成本。在看见收益和认清自己之前，可能大多数人也会像蔡桓公那样默认拒绝吧：“医之好治不病以为功”。成本最低的时候，可能就是开始写第一行产品代码的时候。话虽如此，任何时候都是实现 DevOps 的最佳时机，因为随企业的扩大和代码库的膨胀，成本一定是越来越高的。另外，完全地追求技术上的卓越而忽视成本也不是 DevOps 的推荐做法。读者们在阅读时，也会看到 DevOps 在一些状况下采取的权衡方案。

你希望在一个大家敞开心胸、相互拥抱的环境里共同协作以打造最好的产品，还是守着自己的一亩三分地，与人争辩这是谁的责任，抱怨人们冷漠的同时拒绝其他人的“与你无关”的要求？从本书开始，应用自己获得的知识，并尝试改造这个世界吧！

关于作者

Joakim Verona 是一位擅长持续交付和 DevOps 的咨询师。自 1994 年以来，在系统开发的所有方面他都曾工作过。他积极地在诸如 web 系统、多媒体系统和软硬件混合系统等复杂的多层系统上做出了领导实践者的贡献。自 2004 年以来，他广泛的技能兴趣把他导向了新兴的 DevOps 领域。

Joakim 在林雪平理工学院完成了计算机科学的硕士学位。他也曾作为咨询师工作在各种各样的工业领域中，例如银行和财务、电信、工程、印刷和排版，还有游戏开发。他还对敏捷领域感兴趣，是一位 Scrum 认证的敏捷教练、Scrum 产品负责人并拥有 Java 认证。

我想要谢谢我的妻子，Marie，在写这本书的时候她就是灵感的源泉。我也想谢谢过去数年曾经一起工作过的所有人和公司，让我可以工作在我喜欢的事情上。

关于审稿人

Per Hedman 是一个富有激情的开发者，他也是一个 DevOps 和持续交付的强烈倡导者，他相信应该让开发者对他们自己所写的代码负起责任。

他是一名软件咨询师，在 21 世纪初曾经是一位开发和运维人员。

特别感谢我的妻子和两个女儿，她们让我开心微笑。

Max Manders 是一名运维工程师，在 FanDuel 工作，这家公司是在线 DFS（Daily Fantasy Sports）游戏的领跑者。Max 先前在 Cloudreach 的运营中心工作，这家公司是 AWS Premier 咨询合作伙伴。Max 充分发挥了他的经验和技巧来促进 DevOps 的发展，他也致力于掌握 Ruby 并通过 Chef 和 Puppet 来倡导基础设施及代码。

Max 是 Whisky Web 的共同创立者和组织者，这是一个苏格兰 web 开发和运维社区的大会。当他不写代码或者是研究最新、最棒的监控和运维工具的时候，Max 喜欢威士忌、演奏爵士乐和长号。Max 和他的妻子 Jo 还有他们的猫咪 Ziggy 和 Maggie 生活在爱丁堡。

前言

DevOps 领域在近年来变得流行而普遍。它是那么的流行，以至于很容易忘记在 2008 年以前，当 Patrick Debois 组织起第一个 DevOps 之日大会时，几乎没人曾经听说过该词。

由开发（developers）和运维（operations）组成的 DevOps 这个词，到底意味着什么？为什么它能造成如此巨大的狂热？

本书的任务就是回答这个看起来很简单的问题。

简短的答案就是：DevOps 旨在将不同的社区，比如开发和运维社区，联合起来变成一个更有效率的整体。

这也反映在本书中。它探索了许多在 DevOps 工作中有用的工具，还有那些更加凝聚人们的工具，这些工具比起那些在人之间划清边界的工具来说更令人喜爱。我们用来进行软件开发的流程也是工具，所以将 DevOsp 相关的不同敏捷流派的各个方面包含进来也是很自然的事。

本书也希望做到像标题说的那样，注重实战。让我们在 DevOps 之路上开始旅程吧！

本书主要内容

第 1 章，DevOps 和持续交付简介，涉及了 DevOps 的背景，并介绍它是怎样融入到敏捷开发的广袤世界的。

第 2 章，洞察全局，它会帮助你了解 DevOps 使用的多个系统如何协同工作，组成一个大整体。

第 3 章，DevOps 如何影响架构，描述了软件架构的各个方面，以及当我们以 DevOps 的视角工作时它对我们的意义。

第 4 章，一切皆代码，解释了如何实现一切皆代码。而且，你需要一个地方来存储代码，这个地方就是组织里的源代码管理系统。

第 5 章，构建代码，解释了为何需要系统来构建代码，介绍了这些系统。

第 6 章，测试代码，展示了如果需要及早发布或者经常性发布代码，我们就得对代码的质量有信心。因此我们需要自动化回归测试。

第 7 章，部署代码，展示了当完成了代码的构建和测试，你需要将其部署到服务器上，这样客户就能使用新部署的特性了。

第 8 章，监控代码，涵盖了代码如何通过选择的部署方案来安全地部署到服务器上。你需要监护着它以使其正常工作。

第 9 章，问题跟踪，介绍了处理组织内开发流程的系统，例如问题跟踪软件。在实现敏捷流程时，这样的系统是很重要的帮手。

第 10 章，物联网和 DevOps，描述了 DevOps 如何在物联网的新兴领域帮助我们。

本书的使用要求

本书包含了许多实用例子。为了融会贯通这些例子，你需要一台机器，最好是基于 GNU/Linux 的操作系统，例如 Fedora。

本书的读者

本书面向那些想要承担更大责任，并了解基础设施如何做到构建现代企业的开发者。本书也面向那些想要更好地支持开发者的运维人员。自动化测试的技术人员也是本书的目标受众。

本书主要是包含了许多实例的技术文档，适合那些想要学习实现具体工作代码的人员。尽管如此，前两章的实践性并不强。它们交代了有助于了解其余章节的背景和概览。

约定

在本书中，你将会发现不同的信息类型使用不同的文本样式来区别。这里列出了一些范例和解释：

文本中的代码、数据库表名、文件夹名、文件名、文件扩展名、路径、伪 URL、用户输入和 Twitter 标签以下列形式展示：“在你的本地安装 `git-review`”。

代码段如下所示：

```
private int positiveValue;
void setPositiveValue(int x){
  this.positiveValue=x;
}

int getPositiveValue(){
  return positiveValue;
}
```

命令行的输入输出如下所示：

```
docker run -d -p 4444:4444 --name selenium-hub selenium/hub
```

新术语和**关键词**粗体显示。你在屏幕上看到的词，例如在菜单或者是对话框里，在文本中看起来像是这样：“我们可以通过**修改**按钮改变状态。”

警告或是注意事项像这样展示。

要诀和技巧是这样的。

下载示例代码

你可以从 `http://www.broadview.com.cn` 下载所有已购买的博文视点书籍的示例代码文件。

勘误表

虽然我们已经尽力谨慎地确保内容的准确性，但错误仍然存在。如果你发现了书中的错误，包括正文和代码中的错误，请告诉我们，我们会非常感激。这样，你不仅帮助了其他读者，也帮助我们改进后续的出版。如发现任何勘误，可以在博文视点网站相应图书的页面提交勘误信息。一旦你找到的错误被证实，你提交的信息就会被接受，我们的网站也会发布这些勘误信息。你可以随时浏览图书页面，查看已发布的勘误信息。

目录